Feedback System with reports

A PROJECT REPORT

Submittedby

Gurkirat Singh (23BCS12193)

Aditya Tandon (23BCS13482)

Jaskeerat Singh (23BCS10601)

in partial fulfillment for the award of the degree of

Computer Science

IN

Engineering

Chandigarh University

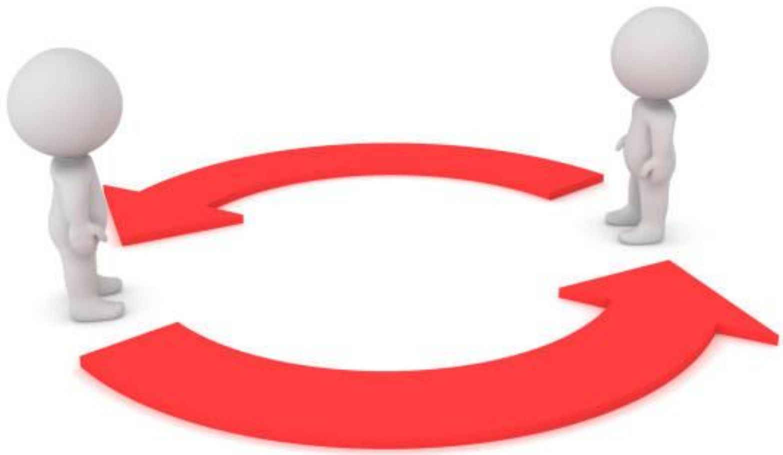
November 2025

TABLE OF CONTENTS

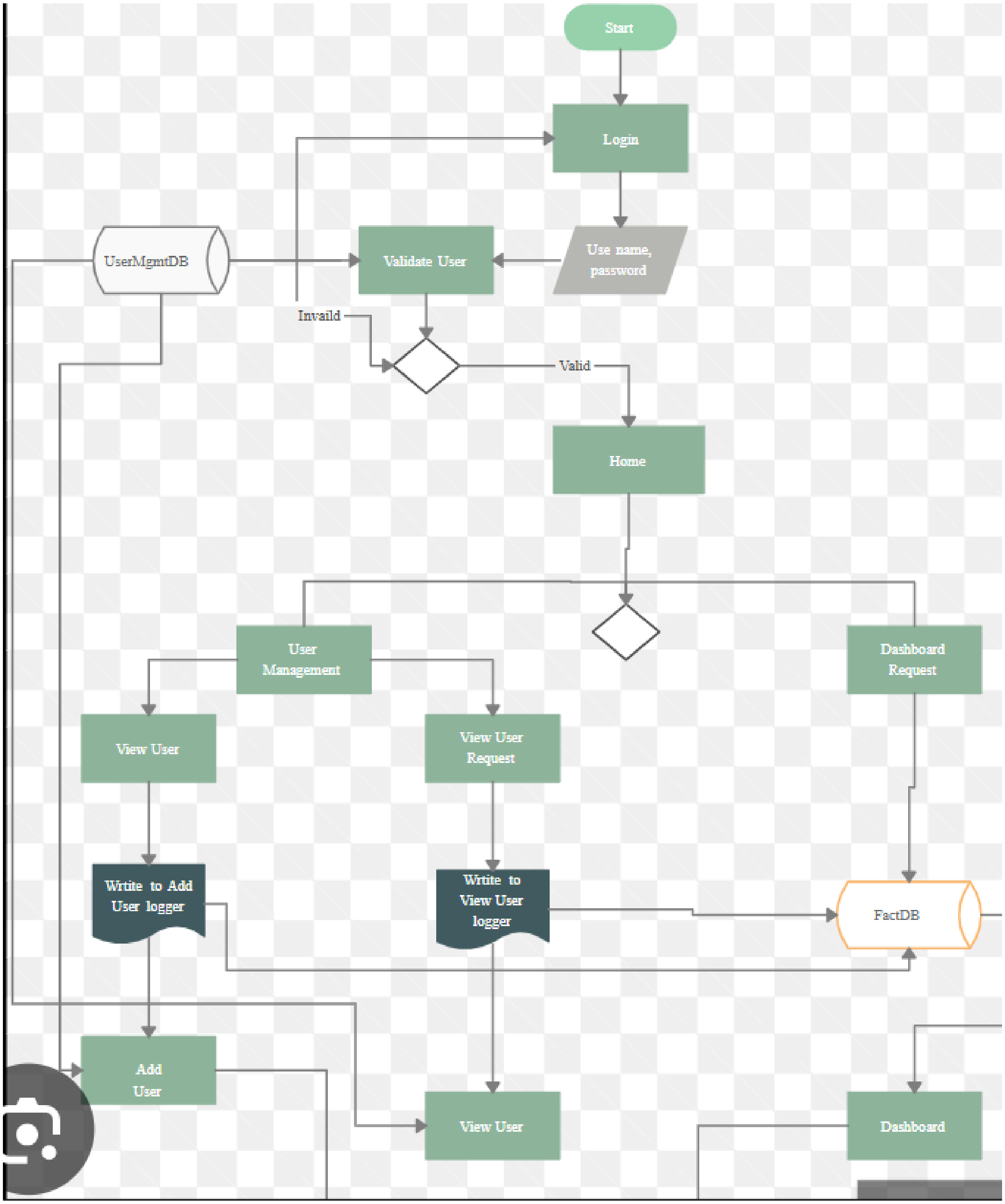
ABSTRACT

The *Feedback System with Reports* is a Java-based application designed to automate and improve the traditional process of collecting academic feedback from students. Conventional paper-based methods are inefficient, time-consuming, and lack analytical capabilities. This system provides a digital platform where students can submit course-wise and faculty-wise feedback using various question types, including rating scales, multiple-choice questions, and open-ended comments. All responses are securely stored in XML format for structured analysis, ensuring data consistency and reliability. The system provides role-based access for students, faculty, and administrators. Faculty can view feedback related to their courses, while administrators can generate institution-wide analytical reports highlighting strengths, weaknesses, and top-performing faculty. The system also provides insights into recurring academic issues and allows administrators to clear all stored data when needed.

The proposed solution improves data accuracy, speeds up analysis, and promotes data-driven decision-making in educational institutions.



GRAPHICAL ABSTRACT



ABBREVIATIONS

Abbreviation	Full Form
XML	Extensible Markup Language
GUI	Graphical User Interface
JVM	Java Virtual Machine
JDK	Java Development Kit
OOP	Object-Oriented Programming

SYMBOLS

Symbol	Description
→	Data Flow
∑	Summation (used in average rating calculation)
%	Percentage rating

CHAPTER 1.

INTRODUCTION

1.1. intro

Educational institutions rely on student feedback to evaluate teaching effectiveness and course quality. However, manual paper-based methods are inefficient, prone to data loss, and difficult to analyze.
A study conducted across various universities reported that over **70%** of institutions still depend on manual methods, leading to delays in report generation and limited actionable insights.
This project addresses the need for a **digital, automated feedback collection and reporting system** to streamline academic evaluation.

1.2. Identification of Problem

The key problem identified is the **inefficiency and lack of analytical capability in traditional feedback systems**. Existing manual processes are time-consuming, unreliable, and do not provide real-time analytics or reports to decision-makers. "Given an array of integers (both positive and negative), find the contiguous subarray that has the largest sum."

The challenge lies in optimizing this process so it runs efficiently, even on large datasets, without excessive time complexity.

1.3. Identification of Tasks

The following tasks were identified:

- Design the system architecture and flow.
- Develop modules for student, faculty, and admin roles.
- Store feedback data in XML format.
- Implement analytics for performance and trend identification.
- Provide report generation and data management features.

1.4. Timeline

Task	Duration (Weeks)
Requirement Analysis	1
System Design	2
Module Implementation	3
XML Storage Integration	1
Report Generation	1
Testing & Validation	1

1.5. Organization of the Report

- **Chapter 1:** Introduction, Problem Identification, and Objectives
- **Chapter 2:** Design Process and Methodology
- **Chapter 3:** Implementation, Results, and Validation
- **Chapter 4:** Conclusion and Future Scope

CHAPTER 2.

DESIGN FLOW/PROCESS

2.1. Evaluation & Selection of Specifications/Features

Required system features:

- Role-based access (Student, Faculty, Admin)
- Secure sign-up and login
- XML-based feedback storage
- Report and analytics generation
- Admin control for data clearing and user management

2.2. Design Constraints

- **Regulatory:** Data privacy compliance

- **Economic:** Cost-effective, open-source tools

- **Technical:** XML file size limitations

- **Ethical:** Anonymity of student responses

- **Safety:** Secure access via password authentication

2.3. Analysis and Feature finalization subject to constraints

After evaluating constraints, the following features were finalized:

- XML-based structured data storage
- Console-based interface (Nimbus-compatible)

2.4. Design Flow

Alternative 1: Web-based system using JSP/Servlets
Alternative 2: Console-based Java application using XML files

2.5. Design selection

Algorithm (Kadane's Algorithm):

The console version was selected due to:

- Simplicity of deployment
- Low resource requirements
- Easy XML integration

2.6. Implementation plan/methodology

```
Start

│

├── Sign Up / Login

│   ├── Student → Submit Feedback → XML Save

│   ├── Faculty → View Feedback Report

│   └── Admin → Generate Reports / Clear Data / View Users

│

├── Analytics → Top Faculty / Average Ratings

│

└── Exit
```

CHAPTER 3.

RESULTS ANALYSIS AND VALIDATION

3.1. Implementation of solution

The system was implemented using **Java (JDK 17)** with XML-based data storage.

Sample Run 1 – Custom Input

```
--- FEEDBACK SYSTEM ---
1. Sign Up
2. Login
3. View Faculty Usernames
4. Analytics & Insights
5. Clear Data (Admin)
6. Exit
```

CHAPTER 4.

CONCLUSION AND FUTURE WORK

4.1. Conclusion

The project successfully developed an **automated feedback management system** that enables structured collection, analysis, and reporting of student feedback.
The XML-based design ensures flexibility and data portability.
Faculty and admins gain valuable insights into teaching effectiveness and institutional performance.

4.2. Future work

- Develop a GUI or Web-based version using JavaFX or JSP.
- Integrate database (MySQL) for large-scale deployments.
- Add graphical report charts (bar/pie charts).
- Enable email notifications and PDF report generation.
- Apply sentiment analysis to detect positive/negative feedback trends.
- Extend the implementation to handle **2D Maximum Subarray Problems** (matrix form).

REFERENCES

- Oracle Java Documentation: https://docs.oracle.com/javase/
- W3C XML Schema Guidelines: https://www.w3.org/XML/
- GeeksforGeeks – XML in Java: https://www.geeksforgeeks.org/xml-parsing-java/
- IEEE Standards for Software Project Reports
- TutorialsPoint – Role-Based Access in Java